M 5

```
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
```

```
                                                              E 16
**FILE**ID**ACTONE




  AAAAAA      CCCCCCCC  TTTTTTTTTT    OOOOOO    NN      NN  EEEEEEEEEE
  AAAAAA      CCCCCCCC  TTTTTTTTTT    OOOOOO    NN      NN  EEEEEEEEEE
AA      AA  CC              TT      OO      OO  NN      NN  EE
AA      AA  CC              TT      OO      OO  NN      NN  EE
AA      AA  CC              TT      OO      OO  NNNN    NN  EE
AA      AA  CC              TT      OO      OO  NNNN    NN  EE
AA      AA  CC              TT      OO      OO  NN  NN  NN  EEEEEEEE
AA      AA  CC              TT      OO      OO  NN  NN  NN  EEEEEEEE
AAAAAAAAAA  CC              TT      OO      OO  NN    NNNN  EE
AAAAAAAAAA  CC              TT      OO      OO  NN    NNNN  EE
AA      AA  CC              TT      OO      OO  NN      NN  EE               ....
AA      AA  CC              TT      OO      OO  NN      NN  EE               ....
AA      AA    CCCCCCCC      TT        OOOOOO    NN      NN  EEEEEEEEEE       ....
AA      AA    CCCCCCCC      TT        OOOOOO    NN      NN  EEEEEEEEEE       ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000
0000            .TITLE  MAC$ACTONE ONCE-ONLY ACTION ROUTINES
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     VAX MACRO ASSEMBLER OBJECT LIBRARY
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000    ; modules for input to the VAX-11 LINKER.
0000    ;
0000    ; ENVIRONMENT:  USER MODE
0000    ;
0000    ; AUTHOR: Benn Schreiber, CREATION DATE: 31-AUG-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-002 MTR0024         Mike Rhodes     4-Feb-1983
0000    ;               Print the full subtitle text in the Table of Contents.
0000    ;
0000    ;       V03.01  MTR0020         Mike Rhodes     7-Jul-1982
0000    ;               Modify handling of enable/disable/show/noshow directives
0000    ;               to also respect command qualifier level CLEARING as well
0000    ;               as SETTING of these options.  Modules affected are
0000    ;               ENABL_DSABL and LIST_NLIST.  A new flag MAC$GL_DSLISF is
0000    ;               employed to facilitate this manipulation.
0000    ;
0000    ;       V01.15  RN0029          R. Newland      10-Mar-1980
0000    ;               Align table of contents so that subtitles containing
0000    ;               tabs remain correctly aligned.
0000    ;
```

```
0000  ;         V01.14  RN0023          R. Newland      2-Nov-1979
0000  ;                 New message codes to get error message from system
0000  ;                 message file.
0000  ;
0000  ;         V01.13  RN0022          R. Newland      31-Oct-1979
0000  ;                 Translate SYS$LP_LINES to set lines/page
0000  ;
0000  ;         V01.12  RN0019          R. Newland      25-Oct-1979
0000  ;                 Improve error pointer positioning
0000  ;
0000  ;         V01.11  RN0005          R. Newland      27-Aug-1979
0000  ;                 Remove .ALIGN LONG statments
0000  ;
0000  ;         V01.12  RN0009          R. Newland      31-Aug-1979
0000  ;                 Allow maximum size IDENT
0000  ;
0000  ;         V01.10  ;RN0002         R. Newland      01-Feb-1979
0000  ;                 Changes for Source Update Merge, .SBTTL line number
0000  ;
0000  ;--
```

```
                                    0000      79          .SBTTL  DECLARATIONS
                                    0000      80  ;
                                    0000      81  ; INCLUDE FILES:
                                    0000      82  ;
                                    0000      83
                                    0000      84  ;
                                    0000      85  ; MACROS:
                                    0000      86  ;
                                    0000      87
                                    0000      88          $RABDEF                                         ;DEFINE RAB OFFSETS
                                    0000      89          $MAC_ADRMODDEF                                  ;DEFINE ADDRESSING MODES
                                    0000      90          $MAC_CTLFLGDEF                                  ;DEFINE CONTROL FLAGS
                                    0000      91          $MAC_INTCODDEF                                  ;DEFINE INT. FILE CODES
                                    0000      92          $MAC_GENVALDEF                                  ;DEFINE COMMON SYMBOLS
                                    0000      93          $MAC_SYMBLKDEF                                  ;DEFINE SYMBOL BLOCK OFFSETS
                                    0000      94          $MACMSGDEF                              ; Define message codes
                                    0000      95          DEFSUMCBL                               ; Define SUM control block symbols
                                    0000      96
                                    0000      97
                                00000000      98          .PSECT  MAC$RO_CODE_P1,NOWRT,GBL,LONG
```

```
                 0000         .SBTTL  ERRORS FOUND BY THE GRAMMAR/PARSER
                 0000
                 0000 ERRENT::                               ;ILLEGAL FORMAT FOR .ENTRY
                 0000         $MAC_ERR BADENTRY              ; Get message code
         31  11  0005         BRB     ERR_0
                 0007
                 0007 ERROPD::                               ;ILLEGAL FORMAT FOR .OPDEF
                 0007         $MAC_ERR ILLOPDEF              ; Get message code
         2A  11  000C         BRB     ERR_0
                 000E
                 000E BDEND1::                               ;.ENDM DIRECTIVE SEEN
                 000E                                        ;OUTSIDE A MACRO DEFINITION
                 000E BDEND2::                               ;.ENDR DIRECTIVE SEEN
                 000E                                        ;OUTSIDE REPEAT BODY
                 000E         $MAC_ERR NOTINMACRO            ; Get message code
         23  11  0013         BRB     ERR_0                  ;ISSUE ERROR, SET CR AND RETURN
                 0015
                 0015 ERRMRS::                               ;MARS_TEXT = ERR01
                 0015         $INTOUT_X INT$_CHKL            ;ALIGN LISTING
                 001B         $MAC_ERR UNRECSTMT             ; Get message code
      FFDD'  30  0020         BSBW    MAC$ERRORPX            ;ISSUE ERROR TO PASS 2
         16  11  0023         BRB     ERR_1                  ;JOIN COMMON CODE
                 0025
                 0025 ERRDOL::                               ;STATEMENT = DIRECTIVE ERR02
                 0025         $MAC_ERR DIRSYNX               ; Get message code
         0C  11  002A         BRB     ERR_0
                 002C
                 002C ERRASN::                               ;ASSIGNMENT = ASSIGN_HEAD ERR02
                 002C         $MAC_ERR ASGNMNTSYN            ; Get message code
         05  11  0031         BRB     ERR_0
                 0033
                 0033 ERRMST::                               ;MACHINE_STAT = MACHINE_INST ERR03
                 0033         $MAC_ERR MCHINSTSYN            ; Get message code
      FFC5'  30  0038 ERR_0:  BSBW    MAC$ERRORPX            ; Report error to pass-2
  5A     0D  9A  003B ERR_1:  MOVZBL  #CR,R10                ;FORCE READING OF NEW LINE
             05  003E         RSB
                 003F
                 003F ERRREF::                               ;OPERANDS = OPERANDS ERR04
                 003F                                        ;REF = ERR06
                 003F         $MAC_ERR OPRNDSYNX             ; Get message code
      FFB9'  30  0044         BSBW    MAC$ERRORPX            ; Issue error to pass-2
      FFB6'  31  0047         BRW     W^MAC$SKP_OPR          ;SKIP TO NEXT OPERAND FIELD
                 004A
                 004A ERRIIF::                               ;IIF_STAT = IIF_HEAD ERR03
                 004A         $MAC_ERR MSGCMAIIF             ; Get message code
         1A  11  004F         BRB     ERR_3                  ;
                 0051
                 0051 ERRDAR::                               ;DATA_LIST = DATA_ARGS ERR04
                 0051                                        ;DATA_ARGS = DATA_LIST DSQOPN EXPR ERR04
                 0051         $MAC_ERR DATALSTSYN            ; Get message code
         13  11  0056         BRB     ERR_3
                 0058
                 0058 ERRADR::                               ;ADDR_LIST = ADDR_LIST ERR04
                 0058                                        ;ADDR_STAT = ADDR_TYPE
                 0058         $MAC_ERR ADRLSTSYNX            ; Get message code
         0C  11  005D         BRB     ERR_3
                 005F
```

```
                         005F  ERRCHA::                                              ;CHAR_ARGS = CHAR_ARGS ERR06
                         005F          $MAC_ERR ILLASCARG                            ; Get message code
                05   11  0064          BRB     ERR_3
                         0066
                         0066  ERRBLK::                                              ;BLOCK_STAT = BLOCK_TYPE ERR03
                         0066          $MAC_ERR BLKDIRSYNX                           ; Get message code
             FF92'   30  006B  ERR_3:  BSBW    MAC$ERRORPT                           ;ISSUE MESSAGE TO PASS 2
                     05  006E          RSB
                         006F
                         006F  ERREXP::                                              ;EXPRESSION ERROR
                         006F          $MAC_ERR ILLEXPR                              ; Get message code
             FF89'   30  0074          BSBW    MAC$ERRORPT                           ;ISSUE ERROR TO PASS 2
             FF86'   31  0077          BRW     MAC$SKP_OPR                           ;SKIP TO NEXT OPERAND FIELD
                         007A
                         007A  ERRBRF::                                              ;BASIC_REF = DOPN ERR01
                         007A                                                        ;BASIC_REF = DAT RRREG
                         007A          $MAC_ERR REGOPSYNX                            ; Get message code
             FF7E'   30  007F          BSBW    MAC$ERRORPT                           ;ISSUE MESSAGE TO PASS 2
 0000'CF    06   90  0082          MOVB    #ADM$_RRIND,W^MAC$GB_MODE  ;SET MODE TO INDIRECT REGISTER
       0000'CF   94  0087          CLRB    W^MAC$GB_REG                          ;USING REGISTER 'R0'
                         008B          $INC_PC                                       ;COUNT ONE BYTE
             FF6E'   31  008F          BRW     MAC$SKP_OPR                           ;SKIP TO CR OR COMMA AND RETURN
```

```
                          0092           .SBTTL  IDENT   PROCESS .IDENT STATEMENT
                          0092
                          0092   ;++
                          0092   ; FUNCTIONAL DESCRIPTION:
                          0092   ;
                          0092   ;       THIS ROUTINE IS CALLED WHEN A .IDENT IS SCANNED.  THE IDENT
                          0092   ;       IS COPIED INTO THE BUFFER MAC$AB_IDENT.
                          0092   ;
                          0092   ;--
                          0092
                          0092   IDENT::
56     0000'CF      9E    0092           MOVAB   W^MAC$AB_IDENT,R6        ;POINT TO IDENT STORAGE
              86    94    0097           CLRB    (R6)+                    ;CLEAR IN CASE NULL IDENT
           FF64'    30    0099           BSBW    MAC$SKIPSP               ;SKIP SPACES
       0D     5A    91    009C           CMPB    R10,#CR                  ;ARE WE AT END OF LINE?
              49    13    009F           BEQL    40$                      ;IF EQL YES
       7E     5A    90    00A1           MOVB    R10,-(SP)                ;NO--SAVE DELIMITER
       55     1F    9A    00A4           MOVZBL  #SYM$K_MAXLEN,R5         ;SET MAX NUMBER OF CHARACTERS
                          00A7   ;
                          00A7   ; LOOP, COLLECTING IDENT. LOOK FOR END OF LINE OR MATCHING DELIMITER
                          00A7   ;
           FF56'    30    00A7   10$:    BSBW    MAC$GETCHR               ;GET NEXT CHARACTER
       6E     5A    91    00AA           CMPB    R10,(SP)                 ;FIND DELIMITER?
              1C    13    00AD           BEQL    20$                      ;IF EQL YES
       0D     5A    91    00AF           CMPB    R10,#CR                  ;NO--END OF LINE?
              17    13    00B2           BEQL    20$                      ;IF EQL YES
    61 8F     5A    91    00B4           CMPB    R10,#^A/A/+^X20          ;IS CHARACTER LOWER CASE?
              09    1F    00B8           BLSSU   15$                      ;IF LSSU NO
    7A 8F     5A    91    00BA           CMPB    R10,#^A/Z/+^X20          ;MAYBE...
              03    1A    00BE           BGTRU   15$                      ;IF GTRU NO
       5A     20    8A    00C0           BICB    #^X20,R10                ;YES--MAKE UPPER CASE
       86     5A    90    00C3   15$:    MOVB    R10,(R6)+                ;NO--STORE CHARACTER
          DE  55    F4    00C6           SOBGEQ  R5,10$                   ; Loop if there is room
              55    D7    00C9           DECL    R5                       ;MAKE R5 NEGATIVE FOR IDENT TOO LONG
50     1F     55    C3    00CB   20$:    SUBL3   R5,#SYM$K_MAXLEN,R0      ;FIGURE LENGTH OF IDENT
0000'CF       50    90    00CF           MOVB    R0,W^MAC$AB_IDENT        ;STORE AS FIRST BYTE OF IDENT
       8E     5A    91    00D4           CMPB    R10,(SP)+                ;END WITH DELIMITER?
              11    13    00D7           BEQL    40$                      ;IF EQL YES
                          00D9           $MAC_ERR UNTERMARG               ; No--assume unterminated arg
              55    D5    00DE           TSTL    R5                       ;BUT CHECK TO SEE
              05    18    00E0           BGEQ    30$                      ;IF GEQ UNTERM. ARG
                          00E2           $MAC_ERR ILLSYMLEN               ; else IDENT is too long
           FF16'    30    00E7   30$:    BSBW    MAC$ERRORPT              ;REPORT ERROR
       5A     0D    9A    00EA   40$:    MOVZBL  #CR,R10                  ;FORCE READING OF NEW LINE
              05          00ED           RSB
```

```
                                        00EE  226         .SBTTL  TITLE   PROCESS .TITLE STATEMENT
                                        00EE  227
                                        00EE  228 ;++
                                        00EE  229 ; FUNCTIONAL DESCRIPTION:
                                        00EE  230 ;
                                        00EE  231 ;       THIS ROUTINE IS CALLED WHEN A .TITLE DIRECTIVE IS SCANNED.
                                        00EE  232 ;       THE REST OF THE SOURCE LINE IS READ AND THE FIRST WORD IS
                                        00EE  233 ;       STORED AS THE PROGRAM TITLE, AND THE REST IS STORED AS THE
                                        00EE  234 ;       TITLE SUB-COMMENT IN THE LISTING HEADER BUFFER.
                                        00EE  235 ;
                                        00EE  236 ;--
                                        00EE  237
                                        00EE  238 TITLE::
                           FF0F'    30  00EE  239         BSBW    MAC$SYMSCNUP            ;SCAN THE SYMBOL
                           5E 50    E9  00F1  240         BLBC    R0,40$                  ;BRANCH IF NO TITLE SCANNED
                  56     0000'CF    9E  00F4  241         MOVAB   W^MAC$AB_TMPSYM,R6      ;POINT TO TEMP SYMBOL NAME BLOCK
                        55    66    9A  00F9  242         MOVZBL  (R6),R5                 ;GET LENGTH OF SYMBOL
                              55    D6  00FC  243         INCL    R5                      ;COPY THE BYTE COUNT ALSO
                  20    66    55    2C  00FE  244         MOVC5   R5,(R6),#^A/ /,-        ;COPY INTO BUFFER WITH BLANK PADDING
                   0000'CF    20        0102  245                 #SYM$K_MAXLEN+1,W^MAC$AB_TITLE ;INTO TITLE BUFFER
                        55    86    9A  0106  246         MOVZBL  (R6)+,R5                ;GET TITLE LENGTH AGAIN
           1F     20    66    55    2C  0109  247         MOVC5   R5,(R6),#^A/ /,#SYM$K_MAXLEN,- ;COPY INTO PAGE HEADER BUFFER
                   0000'CF              010E  248                         W^MAC$AB_HD_TITLE
                           FEEC'    30  0111  249         BSBW    MAC$SKIPSP              ;SKIP SPACES
                  56     0000'CF    9E  0114  250         MOVAB   W^MAC$AB_HD_TSTRG,R6    ;POINT TO WHERE TITLE SUBSTRING GOES
                  20    6E    00    2C  0119  251         MOVC5   #0,(SP),#^A7 /,-        ;BLANK FILL THE TITLE BUFFER
                        66    28        011D  252                       #LST$K_TITLE_SIZ,(R6) ;...
 55  00000001'8F         0000'CF    C1  011F  253         ADDL3   W^MAC$GL_LINELN,#MAC$AB_LINEBF+1,R5 ;COMPUTE LENGTH OF SUBSTRING
                  55     0000'CF    C2  0129  254         SUBL2   W^MAC$GL_LINEPT,R5      ;...
                              22    15  012E  255         BLEQ    40$                     ;IF LEQ NO SUBSTRING
                        28    55    91  0130  256         CMPB    R5,#LST$K_TITLE_SIZ     ;STRING TOO LONG?
                              03    1B  0133  257         BLEQU   10$                     ;IF LEQ NO
                        55    28    9A  0135  258         MOVZBL  #LST$K_TITLE_SIZ,R5     ;YES--USE MAXIMUM SIZE
                   0000'CF    55    D0  0138  259 10$:    MOVL    R5,W^MAC$GL_TTX_SIZ     ;SAVE SUBSTRING LENGTH
           54      0000'CF    01    C3  013D  260         SUBL3   #1,W^MAC$GL_LINEPT,R4   ;POINT TO BEGINNING OF TITLE SUBSTRING
                                        0143  261 ;
                                        0143  262 ; COPY TITLE SUBSTRING INTO PAGE HEADER BUFFER
                                        0143  263 ;
                        66    84    90  0143  264 20$:    MOVB    (R4)+,(R6)              ;COPY A BYTE
                        09    86    91  0146  265         CMPB    (R6)+,#TAB              ;IS CHARACTER A TAB?
                              04    12  0149  266         BNEQ    30$                     ;IF NEQ NO
                     FF A6    20    90  014B  267         MOVB    #^A/ /,-1(R6)           ;YES--MAKE INTO A SPACE
                           F1 55    F5  014F  268 30$:    SOBGTR  R5,20$                  ;DO WHOLE STRING
                        5A    0D    9A  0152  269 40$:    MOVZBL  #CR,R10                 ;FORCE READING OF NEW LINE
                                    05  0155  270         RSB
```

```
                                     0156   272          .SBTTL  SBTTL   PROCESS .SBTTL STATEMENT
                                     0156   273
                                     0156   274 ;++
                                     0156   275 ; FUNCTIONAL DESCRIPTION:
                                     0156   276 ;
                                     0156   277 ;        THIS ROUTINE PROCESSES THE .SBTTL STATEMENT.  THE SUBTITLE
                                     0156   278 ;        LINE IS READ AND WRITTEN TO THE INTERMEDIATE FILE.  IF WE
                                     0156   279 ;        ARE LISTING, THE SUBTITLE LINE IS OUTPUT TO THE LISTING FILE
                                     0156   280 ;
                                     0156   281 ;--
                                     0156   282
                                     0156   283 SBTTL::
                           58    DD  0156   284          PUSHL   R8                          ;PRESERVE R .
    56   0000'CF    00000001'8F  C1  0158   285          ADDL3   #MAC$AB_LINEBF+1,W^MAC$GL_LINELN,R6 ;FIGURE LENGTH OF SUBTITLE
               56   0000'CF      C2  0162   286          SUBL2   W^MAC$GL_LINEPT,R6          ;...
                     58    56    D0  0167   287          MOVL    R6, R8                      ;SAVE LENGTH FOR TABLE OF CONTENTS.
                     28    56    91  016A   288          CMPB    R6,#LST$K_TITLE_SIZ         ;IS SIZE OK?
                           03    1B  016D   289          BLEQU   10$                         ;IF LEQU YES
                     56    28    9A  016F   290          MOVZBL  #LST$K_TITLE_SIZ,R6         ;NO--USE MAXIMUM
               50    59    56    C1  0172   291 10$:     ADDL3   R6,R9,R0                    ;SEE IF THERE IS ENOUGH ROOM IN THE BUFFER
                     50    03    C0  0176   292          ADDL2   #3,R0                       ;COUNT THE 2 COUNT BYTES AND ACTION BYTE
             0000'CF       50    D1  0179   293          CMPL    R0,W^MAC$GL_INTWRNPT        ;ROOM IN THE BUFFER?
                           03    1B  017E   294          BLEQU   20$                         ;IF LEQ YES
                        FE7D'    30  0180   295          BSBW    MAC$OUTFRAME                ;NO--DUMP THE BU;FER
               50    03    56    C1  0183   296 20$:     ADDL3   R6,#3,R0                    ;FIGURE TOTAL LENGTH OF FRAME
                        FE76'    30  0187   297          BSBW    MAC$INTOUT_N                ;SET TO STORE SBTTL IN BUFFER
                     89    21    90  018A   298          MOVB    #INT$_SBTTL,(R9)+           ;STORE THE ACTION CODE
                     89    56    90  018D   299          MOVB    R6,(R9)+                    ;STORE LENGTH OF SBTTL STRING
                           03    14  0190   300          BGTR    40$                         ;IF GTR THERE IS A SUBTITLE STRING
                         00CC    31  0192   301 30$:     BRW     100$                        ;ELSE WE ARE DONE NOW
          55   0000'CF     01    C3  0195   302 40$:     SUBL3   #1,W^MAC$GL_LINEPT,R5       ;POINT TO REAL START OF SBTTL STRING
                           55    DD  019B   303          PUSHL   R5                          ;SAVE OVER MOVC
                           59    DD  019D   304          PUSHL   R9                          ;SAVE POINTER INTO INTERMEDIATE BUFFER
               69    65    56    28  019F   305          MOVC3   R6,(R5),(R9)                ;COPY STRING TO INTERMEDIATE BUFFER
                     59    53    D0  01A3   306          MOVL    R3,R9                       ;UPDATE POINTER INTO INTERMEDIATE BUFFER
                        52 8ED0      01A6   307          POPL    R2                          ;GET POINTER TO START OF SUBTTL IN BUFFER
               62    56    09    3A  01A9   308 50$:     LOCC    #TAB,R6,(R2)                ;FIND ANY TABS IN THE SUBTITLE IN INT. BUFFE
                           05    13  01AD   309          BEQL    60$                         ;IF EQL NO MORE
                     61    20    90  01AF   310          MOVB    #^A/ /,(R1)                 ;FOUND ONE--CHANGE TO SPACE
                           F5    11  01B2   311          BRB     50$                         ;LOOK FOR MORE
                        55 8ED0      01B4   312 60$:     POPL    R5                          ;RESTORE POINTER TO SBTTL STRING
          0000'CF    65    58    28  01B7   313          MOVC3   R8,(R5),W^MAC$AB_LINEBF     ;COPY FOR TABLE OF CONTENTS
                D1 6B      09    E1  01BD   314          BBC     #FLG$V_LSTXST,(R11),30$     ;BRANCH IF NOT DOING LISTINGS
                31 6B      13    E2  01C1   315          BBSS    #FLG$V_TOCFLG,(R11),70$     ;SET TOC FLAG AND SEE IF WE NEED A HEADER
                                     01C5   316 ;
                                     01C5   317 ; OUTPUT TABLE OF CONTENTS HEADER
                                     01C5   318 ;
          0000'CF    0000'CF     D0  01C5   319          MOVL    W^MAC$GL_LN_PAGE,W^MAC$GL_LINE_CNT ; Set # lines left on page
               55    0000'CF     9E  01CC   320          MOVAB   W^MAC$LIST_RAB,R5           ;POINT TO LISTING RAB
            22 A5    0001'8F     B0  01D1   321          MOVW    #MAC$K_HD_SIZE+1,RAB$W_RSZ(R5) ;SET THE RECORD SIZE
            28 A5    0000'CF     9E  01D7   322          MOVAB   W^MAC$AB_HD_NEWPG,RAB$L_RBF(R5) ;AND THE RECORD ADDRESS
                         0088    30  01DD   323          BSBW    SBT_PUT_LIN                 ;WRITE THE TITLE LINE
               50    0000'CF     9E  01E0   324          MOVAB   W^MAC$AB_TOC_MSG,R0         ;POINT TO TABLE OF CONTENTS MSG
                  22 A5    80    9B  01E5   325          MOVZBW  (R0)+,RAB$W_RSZ(R5)         ;SET THE RECORD SIZE
                  28 A5    50    D0  01E9   326          MOVL    R0,RAB$L_RBF(R5)            ;AND THE RECORD ADDRESS
                         0078    30  01ED   327          BSBW    SBT_PUT_LIN                 ;WRITE THE LINE
                        22 A5    B4  01F0   328          CLRW    RAB$W_RSZ(R5)               ;WRITE A BLANK LINE
```

```
              0072   30  01F3           BSBW    SBT_PUT_LIN               ;WRITE THE BLANK LINE
     51   FFE8'CF   9E  01F6  70$:      MOVAB   W^MAC$AB_LINEBF-24,R1     ; Point into listing buffer
81    20202020 8F   D0  01FB            MOVL    #^A/    7,(R1)+           ; Store four spaces
          81   28   90  0202            MOVB    #^A/(/,(R1)+              ; and left parenthesis
     50   0000'CF   D0  0205            MOVL    W^MAC$GL_SRCPAG,R0        ;GET THE SOURCE PAGE NUMBER
             FDF3'  30  020A            BSBW    MAC$DEC_OUT_L2X           ;OUTPUT THE PAGE NUMBER
     80   2029 8F   B0  020D            MOVW    #^A/) /,(R0)+             ;STORE RIGHT PAREN-SPACE
          80   20   90  0212  80$:      MOVB    #^A/ /,(R0)+              ;PAD WITH SPACES
   0000'8F     50   B1  0215            CMPW    R0,#MAC$AB_LINEBF         ;DONE PADDING?
               F6   1F  021A            BLSSU   80$                       ;LOOP FOR ALL
     51   50   08   C3  021C            SUBL3   #8,R0,R1                  ; Form pointer to line number
   0000'CF     51   D0  0220            MOVL    R1,W^MAC$GL_LIST_PTR      ; Set up pointer
     50   0000'CF   3C  0225            MOVZWL  W^MAC$GW_LST_LINE,R0      ; Get listing line number
             FDD3'  30  022A            BSBW    MAC$DEC_OUT_R2L           ;OUTPUT PAGE NUMBER
       11 6B   27   E1  022D            BBC     #FLG$V_OPDFIL,(R11),90$ ; Branch if file is not being updated
   001C'CF     02   E1  0231            BBC     #SUM_V_SRCUPD, -          ; Branch if line is from source
               0B       0236                    W^MAC$GT_SCB+SUM_B_FLAGS,90$
          81   2E   90  0237            MOVB    #^A/./,(R1)+              ; Store period
     50   0000'CF   3C  023A            MOVZWL  W^MAC$GW_LST_INST,R0      ; Get insert number
             FDBE'  30  023F            BSBW    MAC$DEC_OUT_L2X           ; Output the insert number
                        0242  90$:
          58   18   C0  0242            ADDL2   #24,R8                    ; Count the line/page
     55   0000'CF   9E  0245            MOVAB   W^MAC$LIST_RAB,R5         ;POINT TO LISTING RAB
       22 A5   58   D0  024A            MOVL    R8,RAB$W_RSZ(R5)          ;STORE RECORD SIZE
  28 A5   FFE8'CF   9E  024E            MOVAB   W^MAC$AB_LINEBF-24,RAB$L_RBF(R5) ; and the record address
              0011  30  0254            BSBW    SBT_PUT_LIN               ;WRITE THE TOC LINE
          0000'CF   D7  0257            DECL    W^MAC$GL_LINE_CNT         ;ONE LESS LINE TO GO
               04   12  025B            BNEQ    100$                      ;IF NEQ MORE ROOM ON PAGE
       00 6B   13   E5  025D            BBCC    #FLG$V_TOCFLG,(R11),100$          ;ELSE FLAG NEW HEADER NEEDED
               58 8ED0  0261  100$:     POPL    R8                        ;RESTORE R8.
          5A   0D   9A  0264            MOVZBL  #CR,R10                   ;FORCE READING OF NEW LINE
                    05  0267            RSB
                        0268
                        0268  SBT_PUT_LIN:
       13 6B   09   E1  0268            BBC     #FLG$V_LSTXST,(R11),10$ ;BRANCH IF LISTING DISABLED
                        026C            $PUT    RAB=(R5),-                ;WRITE LINE TO LISTING FILE
                        026C                    ERR=W^MAC$ERR_PUT         ;REPORT ANY ERRORS
            03 50   E8  0279            BLBS    R0,10$                    ;BRANCH IF GOOD PUT
             FD81'  30  027C            BSBW    MAC$CLOSE_LIST            ;ELSE CLOSE THE LISTING FILE
                    05  027F  10$:      RSB                               ;KEEP ASSEMBLING
```

```
                           0280              .SBTTL  ENABL/DSABL PROCESS .ENABL/.DSABL
                           0280
                           0280    ;++
                           0280    ; FUNCTIONAL DESCRIPTION:
                           0280    ;
                           0280    ;         THESE TWO ROUTINES PROCESS .ENABL/.DSABL DIRECTIVES.  THE
                           0280    ;         APPROPRIATE FLAGS ARE SET/CLEARED ON PASS 1 AND CODE IS
                           0280    ;         PUT IN THE INTERMEDIATE BUFFER TO DO THE SAME ON PASS 2.
                           0280    ;
                           0280    ;--
                           0280
                           0280    ENABL::                                           ;DIRECTIVE = KENABL
        50    FF 8F    98  0280              CVTBL   #-1,R0                          ;ENABLE ANY OPTIONS
                 02    11  0284              BRB     ENABL_DSABL                     ;GO TO COMMON CODE
                           0286
                           0286    DSABL::                                           ;DIRECTIVE = KDSABL
                 50    D4  0286              CLRL    R0                              ;DISABLE ANY OPTIONS
                           0288
                           0288    ENABL_DSABL:
                           0288
  0000'CF        50    D0  0288              MOVL    R0,W^MAC$GL_DIRFLG              ;SET THE FLAG FOR LATER
              FD70'    30  028D              BSBW    MAC$SYMSCNUP                    ;SCAN NEXT OPTION
              67 50    E9  0290              BLBC    R0,50$                          ;BRANCH IF NO OPTION SCANNED
     55     0000'CF    9E  0293              MOVAB   W^ENB$G_OPTIONS,R5              ;POINT TO ENABLE OPTIONS LIST
              FD65'    30  0298              BSBW    MAC$SRC_LIST                    ;LOOK FOR THE OPTION WE SCANNED
              1B 50    E8  029B              BLBS    R0,20$                          ;BRANCH IF FOUND
     55     0000'CF    9E  029E              MOVAB   W^ENB$G_LONGNAMES,R5            ;NO--TRY THE LONG NAMES
              FD5A'    30  02A3              BSBW    MAC$SRC_LIST                    ;LOOK FOR IT
              06 50    E9  02A6              BLBC    R0,10$                          ;BRANCH IF NOT FOUND
        51    05 A1    D0  02A9              MOVL    SYM$L_VAL(R1),R1                ;POINT TO THE REAL BLOCK
                 0A    11  02AD              BRB     20$                             ;AND CONTINUE
                           02AF    10$:      $MAC_ERR NOTENABOPT                     ; Get error message code
              FD49'    30  02B4              BSBW    MAC$ERRORLN                     ;SEND TO PASS 2
                 34    11  02B7              BRB     40$                             ;
  0000'8F        51    B1  02B9    20$:      CMPW    R1,#ENB$G_LOCALSYMB             ;WAS THIS '.ENABL LSB'?
                 08    12  02BE              BNEQ    30$                             ;IF NEQ NO
     03 0000'CF        E9  02C0              BLBC    W^MAC$GL_DIRFLG,30$             ;YES--BRANCH IF DISABLE
              FD38'    30  02C5              BSBW    MAC$SET_NEW_LSB                 ;NO--ENABLE A NEW LSB
0000'CF       09 A1    B3  02C8    30$:      BITW    SYM$W_FLAG(R1),W^MAC$GL_ENLISF ;SET BY COMMAND?
                 1D    12  02CE              BNEQ    40$                             ;IF NEQ YES--DO NOT CHANGE HERE
0000'CF       09 A1    B3  02D0              BITW    SYM$W_FLAG(R1),W^MAC$GL_DSLISF ; CLEARED BY COMMAND?
                 15    12  02D6              BNEQ    40$                             ;IF NEQ YES--DO NOT CHANGE HERE
     52       05 A1    9E  02D8              MOVAB   SYM$L_VAL(R1),R2                ;POINT TO THE VALUE WORD FOR OPTION
   62       0000'CF    D0  02DC              MOVL    W^MAC$GL_DIRFLG,(R2)            ;SET/CLEAR THE OPTION
                           02E1              $INTOUT_LW INT$_SETLONG,<W^MAC$GL_DIRFLG,R2> ;SET/CLEAR FLAG ON PASS 2
              FD10'    30  02ED    40$:      BSBW    MAC$SKIPSP                      ;SKIP SPACES
          2C     5A    91  02F0              CMPB    R10,#^A/,/                      ;SCAN TO A COMMA?
                 93    12  02F3              BNEQ    ENABL_DSABL                     ;IF NEQ NO--SCAN FOR NEXT OPTION
              FD08'    30  02F5              BSBW    MAC$GETCHR                      ;YES--SKIP THE COMMA
                 F3    11  02F8              BRB     40$                             ;CONTINUE
                       05  02FA    50$:      RSB
```

```
                         02FB          .SBTTL  LIST/NLIST PROCESS .LIST/.NLIST
                         02FB
                         02FB   ;++
                         02FB   ; FUNCTIONAL DESCRIPTION:
                         02FB   ;
                         02FB   ;       THESE TWO ROUTINES PROCESS THE .LIST/.NLIST DIRECTIVES.
                         02FB   ;       THE LINE IS SCANNED TO GET THE OPTIONS (IF ANY) AND THE
                         02FB   ;       APPROPRIATE FLAGS ARE CLEARED IN PASS 1 AND CODE IS SENT
                         02FB   ;       TO THE INTERMEDIATE BUFFER TO DO SO ON PASS 2.
                         02FB   ;
                         02FB   ;--
                         02FB
                         02FB   LIST::                                          ;DIRECTIVE = KLIST
         50   FF 8F  98  02FB           CVTBL   #-1,R0                          ;SET ANY FLAGS
                 01  DD  02FF           PUSHL   #1                              ;INCREMENT LISTING LEVEL
                 06  11  0301           BRB     LIST_NLIST
                         0303
                         0303   NLIST::                                         ;DIRECTIVE = KNLIST
                 50  D4  0303           CLRL    R0                              ;CLEAR ANY FLAGS
         7E   FF 8F  98  0305           CVTBL   #-1,-(SP)                       ;DECREMENT LEVEL
                         0309
                         0309   LIST_NLIST:
                         0309
    0000'CF      50  D0  0309           MOVL    R0,W^MAC$GL_DIRFLG              ;SET THE FLAG FOR LATER
            FCEF'    30  030E           BSBW    MAC$SYMSCNUP                    ;SCAN FOR AN OPTION
              24 50  E8  0311           BLBS    R0,10$                          ;BRANCH IF OPTION SCANNED
    0000'CF      8E  C0  0314           ADDL2   (SP)+,W^MAC$GL_LIST_LVL ;NO--ADJUST LISTING LEVEL
                         0319           $INTOUT_LW INT$_SETLONG,-       ;SEND CODE FOR PASS 2 TO DO THE SAME
                         0319                   <W^MAC$GL_LIST_LVL,#MAC$GL_LIST_LVL>
                         0329           $INTOUT_LW INT$_SETLONG,<#1,#MAC$GL_LIST_IT> ;SET '.LIST/.NLIST' FLAG
                     05  0337           RSB                                     ;ALL DONE
                         0338   ;
                         0338   ; THERE WAS AT LEAST ONE OPTION ON THE LINE
                         0338   ;
                 8E  D5  0338   10$:    TSTL    (SP)+                           ;CLEAR THE STACK
    55     0000'CF   9E  033A   20$:    MOVAB   W^LST$G_DIRLIST,R5              ;POINT TO OPTION NAMES
            FCBE'    30  033F           BSBW    MAC$SRC_LIST                    ;LOOK UP THE OPTION NAME
              1B 50  E8  0342           BLBS    R0,40$                          ;BRANCH IF FOUND
    55     0000'CF   9E  0345           MOVAB   W^LST$G_LONGNAMES,R5            ;NO--TRY THE LONG NAMES
            FCB3'    30  034A           BSBW    MAC$SRC_LIST                    ;LOOK FOR IT
              06 50  E9  034D           BLBC    R0,30$                          ;BRANCH IF NOT FOUND
         51   05 A1  D0  0350           MOVL    SYM$L_VAL(R1),R1                ;POINT TO THE REAL BLOCK
                 0A  11  0354           BRB     40$                             ;AND CONTINUE
                         0356   30$:    $MAC_ERR NOTLGLISOP                     ; Get message code
            FCA2'    30  035B           BSBW    MAC$ERRORLN                     ;SEND TO PASS 2
                 23  11  035E           BRB     50$                             ;
                         0360   ;
                         0360   ; OPTION WAS FOUND
                         0360   ;
 0000'CF      09 A1  B3  0360   40$:    BITW    SYM$W_FLAG(R1),W^MAC$GL_ENLISF ;SET BY COMMAND?
                 1B  12  0366           BNEQ    50$                             ;YES--DO NOT CHANGE HERE
 0000'CF      09 A1  B3  0368           BITW    SYM$W_FLAG(R1),W^MAC$GL_DSLISF ; CLEARED BY COMMAND?
                 13  12  036E           BNEQ    50$                             ;IF NEQ YES--DO NOT CHANGE HERE
       52     05 A1  9E  0370           MOVAB   SYM$L_VAL(R1),R2                ;POINT TO VALUE
     62     0000'CF  D0  0374           MOVL    W^MAC$GL_DIRFLG,(R2)            ;SET/CLEAR FLAG
                         0379           $INTOUT_LW INT$_SETLONG,<(R2),R2> ;TELL PASS 2 TO DO IT TOO
            FC7A'    30  0383   50$:    BSBW    MAC$SKIPSP                      ;SKIP SPACES
```

```
      2C    5A  91  0386   479          CMPB    R10,#^A/,/              ;DID WE GET TO A COMMA?
            05  12  0389   480          BNEQ    60$                     ;IF NEQ NO
         FC72'  30  038B   481          BSBW    MAC$GETCHR              ;YES--GET NEXT CHARACTER
            F3  11  038E   482          BRB     50$                     ;
         FC6D'  30  0390   483 60$:     BSBW    MAC$SYMSCNUP            ;FIND AN OPTION
         A4 50  E8  0393   484          BLBS    R0,20$                  ;BRANCH IF OPTION SCANNED
                05  0396   485          RSB
```

```
                        0397            .SBTTL  PROCESS .CROSS/.NOCROSS DIRECTIVES
                        0397
                        0397    ;++
                        0397    ; FUNCTIONAL DESCRIPTION:
                        0397    ;
                        0397    ;       THESE TWO ROUTINES PROCESS THE .CROSS AND .NOCROSS
                        0397    ;       DIRECTIVES.
                        0397    ;
                        0397    ;--
                        0397
                        0397    CROENB::                                        ;DIRECTIVE = KCROSS
       03F2'CF      9F  0397            PUSHAB  W^CLR_XCRF                      ;ACTION ROUTINE FOR SYMBOLS
            56      D4  039B            CLRL    R6                              ;FLAG THIS IS ENABLE
            07      11  039D            BRB     CROS_0                          ;JOIN COMMON CODE
                        039F
                        039F    CRODSB::                                        ;DIRECTIVE = KNCROS
       03EC'CF      9F  039F            PUSHAB  W^SET_XCRF                      ;ACTION ROUTINE FOR SYMBOLS
       56   01      D0  03A3            MOVL    #1,R6                           ;FLAG THIS IS DISABLE
         FC57'      30  03A6    CROS_0: BSBW    MAC$SKIPSP                      ;SKIP SPACES
       0D   5A      91  03A9            CMPB    R10,#CR                         ;JUST THE DIRECTIVE?
            2F      13  03AC            BEQL    70$                             ;IF EQL YES
                        03AE    ;
                        03AE    ; LOOP, SCANNING SYMBOL NAMES.  SET OR CLEAR SYM$M_XCRF IN THE
                        03AE    ; SYMBOL FLAGS FOR EACH SYMBOL FOUND, AS APPROPRIATE.
                        03AE    ;
         FC4F'      30  03AE    20$:    BSBW    MAC$SYMSCNUP                    ;SCAN A SYMBOL NAME
         0A 50      E8  03B1            BLBS    R0,30$                          ;BRANCH IF WE GOT ONE
                        03B4            $MAC_ERR DIRSYNX                        ; No--syntax error
         FC44'      30  03B9            BSBW    MAC$ERRORPT                     ;REPORT THE ERROR
            1C      11  03BC            BRB     60$                             ;FINISH UP
         FC3F'      30  03BE    30$:    BSBW    MAC$INSUSRSYMTB                 ;LOOK UP THE SYMBOL NAME (INSRT IF NOT FOUND
         03 50      E9  03C1            BLBC    R0,40$                          ;IF NOT FOUND, JUST IGNORE IT
         00 BE      16  03C4            JSB     @(SP)                           ;CALL ACTION ROUTINE TO SET/CLEAR XCRF
         FC36'      30  03C7    40$:    BSBW    MAC$SKIPSP                      ;SKIP SPACES
       2C   5A      91  03CA            CMPB    R10,#^A/,/                      ;STOP ON A COMMA?
            06      12  03CD            BNEQ    50$                             ;IF NEQ NO
         FC2E'      30  03CF            BSBW    MAC$GETCHR                      ;YES--GET NEXT CHAR
         FC2B'      30  03D2            BSBW    MAC$SKIPSP                      ;AND THEN SKIP SPACES
       0D   5A      91  03D5    50$:    CMPB    R10,#CR                         ;END OF LINE?
            D4      12  03D8            BNEQ    20$                             ;IF NEQ NO
            8E      D5  03DA    60$:    TSTL    (SP)+                           ;YES--CLEAR STACK
                    05  03DC            RSB
                        03DD    ;
                        03DD    ; THERE WERE NO SYMBOL NAMES ON THE LINE.  IF .CROSS, CLEAR THE XCRF
                        03DD    ; BIT IN FLAGS.   IF .NOCROSS, SET IT.
                        03DD    ;
            8E      D5  03DD    70$:    TSTL    (SP)+                           ;CLEAR THE STACK
         05 56      E9  03DF            BLBC    R6,90$                          ;BRANCH IF THIS WAS .CROSS
    00 6B   1F      E3  03E2            BBCS    #FLG$V_XCRF,(R11),.+1   ;.NOCROSS--SET XCRF FLAG
                    05  03E6            RSB                                     ;ALL DONE
    00 6B   1F      E5  03E7    90$:    BBCC    #FLG$V_XCRF,(R11),.+1   ;.CROSS--CLEAR XCRF FLAG
                    05  03EB            RSB
                        03EC
                        03EC            .DEBUG  SET_XCRF, CLR_XCRF
                        03EC
                        03EC    SET_XCRF:
 00 09 A1   0C      E3  03EC            BBCS    #SYM$V_XCRF,SYM$W_FLAG(R1),.+1 ;DISABLE CREF FOR THIS SYMBOL
```

```
                           05  03F1    544            RSB
                               03F2    545
                               03F2    546 CLR_XCRF:
      00 09 A1     0C      E5  03F2    547            BBCC    #SYM$V_XCRF,SYM$W_FLAG(R1),.+1 ;ENABLE CREF FOR THIS YSMBOL
                           05  03F7    548            RSB
```

```
                         03F8                 .SBTTL  SETDFL  PROCESS .DEFAULT DIRECTIVE
                         03F8
                         03F8      ;++
                         03F8      ; FUNCTIONAL DESCRIPTION:
                         03F8      ;
                         03F8      ;       THIS ROUTINE PROCESSES THE .DEFAULT DIRECTIVE.  THE ONLY
                         03F8      ;       ARGUMENT CURRENTLY IMPLEMENTED IS ''DISPLACEMENT''.  THIS
                         03F8      ;       SETS THE DEFAULT DISPLACEMENT TO USE IN PC-RELATIVE EXPRESSIONS
                         03F8      ;       WHEN NO EXPLICIT DISPLACEMENT IS SPECIFIED.
                         03F8      ;
                         03F8      ;--
                         03F8
                         03F8      SETDFL::                                            ;DIRECTIVE = KDFLT
            FC05'   30   03F8              BSBW    MAC$SYMSCNUP                        ;GET THE THING WE ARE DEFAULTING
          07 50     E8   03FB              BLBS    R0,10$                              ;BRANCH IF WE FOUND SOMETHING
                         03FE              $MAC_ERR DIRSYNX                            ; No symbol--that's an erroR
              52    11   0403              BRB     60$                                 ;REPORT ERROR AND RETURN
    55   0000'CF    9E   0405      10$:    MOVAB   W^MAC$DFLT_LIST,R5                  ;POINT TO THE LIST OF VALID ARGS
            FBF3'   30   040A              BSBW    MAC$SRC_LIST                        ;LOOK UP THE ARG
          07 50     E8   040D              BLBS    R0,20$                              ;BRANCH IF FOUND
                         0410              $MAC_ERR ILLDFLTARG                         ; No--report the error
              40    11   0415              BRB     60$                                 ;AND RETURN
                         0417      ;
                         0417      ; THERE IS CURRENTLY ONLY ONE THING DEFAULTED.  HENCE, IT IS
                         0417      ; NOT CHECKED.
                         0417      ;
            FBE6'   30   0417      20$:    BSBW    MAC$SKIPSP                          ;SKIP SPACES
         2C   5A    91   041A              CMPB    R10,#^A/,/                          ;SKIP TO A COMMA?
              06    12   041D              BNEQ    30$                                 ;IF NEQ NO
            FBDE'   30   041F              BSBW    MAC$GETCHR                          ;YES--SKIP IT
            FBDB'   30   0422              BSBW    MAC$SKIPSP                          ;THEN SKIP SPACES
            FBD8'   30   0425      30$:    BSBW    MAC$SYMSCNUP                        ;SCAN FOR ANOTHER SYMBOL
          07 50     E8   0428              BLBS    R0,40$                              ;BRANCH IF WE FOUND ONE
                         042B              $MAC_ERR DIRSYNX                            ; No--syntax error
              25    11   0430              BRB     60$                                 ;REPORT ERROR
    55   0000'CF    9E   0432      40$:    MOVAB   W^MAC$DSPL_ARGS,R5                  ;POINT TO DISPLACEMENT ARGS LIST
            FBC6'   30   0437              BSBW    MAC$SRC_LIST                        ;LOOK UP THE OPTION
          07 50     E8   043A              BLBS    R0,50$                              ;BRANCH IF FOUND
                         043D              $MAC_ERR ILLDFLTARG                         ; No--illegal arg
              13    11   0442              BRB     60$                                 ;REPORT ERROR
0000'CF    05 A1    D0   0444      50$:    MOVL    SYM$L_VAL(R1),W^MAC$GL_DFPC_DSP ;SET DEFAULT DISPLACEMENT
            FBB3'   30   044A              BSBW    MAC$SKIPSP                          ;NOW SKIP SPACES
         0D   5A    91   044D              CMPB    R10,#CR                             ;MAKE SURE WE FOUND END OF LINE
              0B    13   0450              BEQL    70$                                 ;IF EQL WE FOUND IT
                         0452              $MAC_ERR DIRSYNX                            ;OOPS--AND WE WERE DOING SO WELL
            FBA6'   30   0457      60$:    BSBW    MAC$ERRORLN                         ;REPORT THE ERROR
         5A   0D    D0   045A              MOVL    #CR,R10                             ;FORCE END OF LINE
                    05   045D      70$:    RSB
```

```
                     045E           .SBTTL  ENDPRG  PROCESS .END STATEMENT
                     045E
                     045E   ;++
                     045E   ; FUNCTIONAL DESCRIPTION:
                     045E   ;
                     045E   ;       THIS ROUTINE IS CALLED WHEN THE '.END' STATEMENT IS ENCOUNTERED.
                     045E   ;       IT WILL SCAN FOR A TRANSFER ADDRESS AND STORE IT AWAY IF IT
                     045E   ;       IS PRESENT.  A JUMP IS THEN EXECUTED TO END PASS 1.
                     045E   ;
                     045E   ;--
                     045E
                     045E   ENDPRG::                                        ;DIRECTIVE = KEND
          FB9F'  30  045E           BSBW    MAC$SYMSCNUP                    ;SCAN FOR A SYMBOL NAME
          0C 50  E8  0461           BLBS    R0,10$                          ;BRANCH IF SYMBOL SCANNED
     0D      5A  91  0464           CMPB    R10,#CR                         ;NO--BUT DID WE FIND END OF LINE?
             1C  13  0467           BEQL    40$                             ;IF EQL YES--OK
                     0469           $MAC_ERR DIRSYNX                        ; Else issue syntax error
             0B  11  046E           BRB     20$                             ;ISSUE MESSAGE
          FB8D'  30  0470   10$:    BSBW    MAC$SRCUSRSYMTB                 ;YES--LOOK UP NAME
          0A 50  E8  0473           BLBS    R0,30$                          ;IF LBS THEN NAME FOUND OK
                     0476           $MAC_ERR UNDEFXFRAD                     ; Undefined--set to issue message
          FB82'  30  0478   20$:    BSBW    MAC$ERRORLN                     ;ISSUE MESSAGE TO PASS 2
             05  11  047E           BRB     40$
  0000'CF    51  D0  0480   30$:    MOVL    R1,W^MAC$GL_XFRADR              ;SET POINTER TO SYMBOL BLOCK
       0000'CF   D5  0485   40$:    TSTL    W^MAC$GL_IF_LEVEL               ;ANY OUTSTANDING CONDITIONALS?
             08  15  0489           BLEQ    50$                             ;IF LEQ NO
                     048B           $MAC_ERR UNTERMCOND                     ; Yes--get message code
          FB6D'  30  0490           BSBW    MAC$ERRORLN                     ;SEND MESSAGE TO PASS 2
          FB6A'  30  0493   50$:    BSBW    MAC$SET_PC                      ;RECORD PC
          FB67'  31  0496           BRW     W^MAC$PASS1_END                 ;FINISH PASS 1
                     0499
                     0499           .END
```

```
$$.TMP1             = 00000002                ERRASN                0000002C RG      03
$$.TMP2             = 00000065                ERRBLK                00000066 RG      03
$COUNT              = 0000003B                ERRBRF                0000007A RG      03
ADM$_ABSOLUTE       = 00000002                ERRCHA                0000005F RG      03
ADM$_BYTE_DISP      = 0000000A                ERRDAR                00000051 RG      03
ADM$_DFBYTEDISP     = 0000000B                ERRDOL                00000025 RG      03
ADM$_DFLONGDISP     = 0000000F                ERRENT                00000000 RG      03
ADM$_DFRAUTOINC     = 00000009                ERREXP                0000006F RG      03
ADM$_DFWORDDISP     = 0000000D                ERRIIF                0000004A RG      03
ADM$_IMMEDIATE      = 00000001                ERRMRS                00000015 RG      03
ADM$_INDEX          = 00000004                ERRMST                00000033 RG      03
ADM$_LITERAL        = 00000000                ERROPD                00000007 RG      03
ADM$_LONG_DISP      = 0000000E                ERRREF                0000003F RG      03
ADM$_MAXMOD         = 0000000F                ERR_0                 00000038 R       03
ADM$_PIC            = 00000003                ERR_1                 0000003B R       03
ADM$_REGAUTODEC     = 00000007                ERR_3                 0000006B R       03
ADM$_REGAUTOINC     = 00000008                FF                  = 0000000C
ADM$_REGISTER       = 00000005                FLG$M_ALLCHR        = 00000001
ADM$_RRIND          = 00000006                FLG$M_BOL           = 00000002
ADM$_WORD_DISP      = 0000000C                FLG$M_CHKLPND       = 00100000
ARG$K_SIZE          = 000003E8                FLG$M_COMPEXPR      = 00000004
AUD$K_SIZE          = 00000010                FLG$M_CONT          = 00000008
BDEND1                0000000E RG      03     FLG$M_CRF           = 40000000
BDEND2                0000000E RG      03     FLG$M_CRSEEN        = 00000001
BIT...              = 00000005                FLG$M_DATRPT        = 00000010
BLNK                = 00000020                FLG$M_DBGOUT        = 00004000
CHR$M_COMMA_CR      = 00000020                FLG$M_DLIMSTR       = 00008000
CHR$M_ILL_CHR       = 00000040                FLG$M_ENDMCH        = 00000020
CHR$M_NUM_BER       = 00000010                FLG$M_EVALEXPR      = 00000040
CHR$M_SPA_MSK       = 00000001                FLG$M_EXPOPT        = 00000080
CHR$M_SYM_CH1       = 00000008                FLG$M_EXTERR        = 00010000
CHR$M_SYM_CHR       = 00000004                FLG$M_EXTWRN        = 00020000
CHR$M_SYM_DLM       = 00000002                FLG$M_FIRSTLN       = 00000200
CHR$V_COMMA_CR      = 00000005                FLG$M_IFSTAT        = 00800000
CHR$V_CVTLWC        = 00000061                FLG$M_IIF           = 00400000
CHR$V_ILL_CHR       = 00000006                FLG$M_INSERT        = 00000100
CHR$V_NOCVT         = 0000007F                FLG$M_IRPC          = 20000000
CHR$V_NUM_BER       = 00000004                FLG$M_LEXOP         = 00000002
CHR$V_SPA_MSK       = 00000000                FLG$M_LSTXST        = 00000200
CHR$V_SYM_CH1       = 00000003                FLG$M_MAC2COL       = 00000800
CHR$V_SYM_CHR       = 00000002                FLG$M_MACL          = 00000800
CHR$V_SYM_DLM       = 00000001                FLG$M_MACLTB        = 08000000
CLR_XCRF              000003F2 R   D   03     FLG$M_MACTXT        = 00010000
CNT                 = 00000002                FLG$M_MEBLST        = 00001000
CR                  = 0000000D                FLG$M_MOREARG       = 00002000
CRODSB                0000039F RG      03     FLG$M_MOREINP       = 00000008
CROENB                00000397 RG      03     FLG$M_NEWPND        = 00000400
CROS_D                000003A6 R       03     FLG$M_NOREF         = 01000000
DSABL                 00000286 RG      03     FLG$M_NTYPEPC       = 00000020
ENABL                 00000280 RG      03     FLG$M_NULCHR        = 00040000
ENABL_DSABL           00000288 R       03     FLG$M_OBJXST        = 00200000
ENB$G_LOCALSYMB       ********      X  03     FLG$M_OPNDCHK       = 00000100
ENB$G_LONGNAMES       ********      X  03     FLG$M_OPRND         = 00002000
ENB$G_OPTIONS         ********      X  03     FLG$M_OPTVFLIDX     = 00001000
ENDPRG                0000045E RG      03     FLG$M_ORDLST        = 00020000
ERR                 = 00000000                FLG$M_P2            = 00004000
ERRADR                00000058 RG      03     FLG$M_RPTIRP        = 10000000
```

```
MAC
Sym

AB
AD
AF
AG
AH
AL
AO
AQ
ARG
AUD
AW
B
BLN
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CR
D
DAN
DAN
DAN
DAT
DBU
DCL
DCO
DCO
DDI
DEO
DEQ
DFN
DGU
DIN
DIU
DLU
DMA
DMI
DOP
DOP
DOR
DPC
DPL
DPO
DSQ
DSQ
```

```
FLG$M_SEQFIL                     = 02000000          FLG$V_TOCFLG                     = 00000013
FLG$M_SKAN                       = 00008000          FLG$V_UPAFLG                     = 00000024
FLG$M_SPECOP                     = 00000004          FLG$V_UPDFIL                     = 00000027
FLG$M_SPLALL                     = 04000000          FLG$V_UPMARG                     = 00000026
FLG$M_STOIMF                     = 00040000          FLG$V_XCRF                       = 0000001F
FLG$M_SYM2COL                    = 00000400          HASHSZ                           = 0000007F
FLG$M_TOCFLG                     = 00080000          HYPHEN                           = 0000002D
FLG$M_UPAFLG                     = 00000010          IDENT                              00000092 RG    03
FLG$M_UPDFIL                     = 00000080          INP$K_BUFSIZ                     = 000003E8
FLG$M_UPMARG                     = 00000040          INT$K_BUFSIZ                     = 000013F4
FLG$M_XCRF                       = 80000000          INT$K_BUFWRN                     = 00001390
FLG$V_ALLCHR                     = 00000000          INT$_ADD                         = 00000001
FLG$V_BOL                        = 00000001          INT$_AND                         = 00000002
FLG$V_CHKLPND                    = 00000014          INT$_ASH                         = 00000003
FLG$V_COMPEXPR                   = 00000002          INT$_ASN                         = 0000000C
FLG$V_CONT                       = 00000003          INT$_AUGPC                       = 0000000D
FLG$V_CRF                        = 0000001E          INT$_BDST                        = 0000000E
FLG$V_CRSEEN                     = 00000020          INT$_CHKL                        = 0000000F
FLG$V_DATRPT                     = 00000004          INT$_DIV                         = 00000004
FLG$V_DBGOUT                     = 0000002E          INT$_END                         = 00000010
FLG$V_DLIMSTR                    = 0000002F          INT$_EPT                         = 00000011
FLG$V_ENDMCH                     = 00000005          INT$_ERR                         = 00000012
FLG$V_EVALEXPR                   = 00000006          INT$_ETX                         = 00000013
FLG$V_EXPOPT                     = 00000007          INT$_FNEWL                       = 00000014
FLG$V_EXTERR                     = 00000030          INT$_ILG                         = 00000000
FLG$V_EXTWRN                     = 00000031          INT$_INFO                        = 0000003A
FLG$V_FIRSTLN                    = 00000029          INT$_LGLAB                       = 00000015
FLG$V_IFSTAT                     = 00000017          INT$_MACL                        = 00000016
FLG$V_IIF                        = 00000016          INT$_MUL                         = 00000005
FLG$V_INSERT                     = 00000008          INT$_NEG                         = 00000006
FLG$V_IRPC                       = 0000001D          INT$_NEWL                        = 00000017
FLG$V_LEXOP                      = 00000021          INT$_NEWP                        = 00000018
FLG$V_LSTXST                     = 00000009          INT$_NOT                         = 00000007
FLG$V_MAC2COL                    = 0000002B          INT$_OP                          = 00000019
FLG$V_MACL                       = 0000000B          INT$_OR                          = 00000008
FLG$V_MACLTB                     = 0000001B          INT$_PRIL                        = 0000001A
FLG$V_MACTXT                     = 00000010          INT$_PRT                         = 0000001B
FLG$V_MEBLST                     = 0000000C          INT$_PSECT                       = 0000001C
FLG$V_MOREARG                    = 0000002D          INT$_REDEF                       = 0000001D
FLG$V_MOREINP                    = 00000023          INT$_REF                         = 0000001E
FLG$V_NEWPND                     = 0000000A          INT$_REST                        = 0000001F
FLG$V_NOREF                      = 00000018          INT$_SAME                        = 00000009
FLG$V_NTYPEPC                    = 00000025          INT$_SAVE                        = 00000020
FLG$V_NULCHR                     = 00000032          INT$_SBTTL                       = 00000021
FLG$V_OBJXST                     = 00000015          INT$_SETFLAG                     = 00000022
FLG$V_OPNDCHK                    = 00000028          INT$_SETLONG                     = 00000023
FLG$V_OPRND                      = 0000000D          INT$_SPIC                        = 00000024
FLG$V_OPTVFLIDX                  = 0000002C          INT$_SPID                        = 00000025
FLG$V_ORDLST                     = 00000011          INT$_STIB                        = 00000026
FLG$V_P2                         = 0000000E          INT$_STIL                        = 00000028
FLG$V_RPTIRP                     = 0000001C          INT$_STIW                        = 00000027
FLG$V_SEQFIL                     = 00000019          INT$_STKEPT                      = 00000029
FLG$V_SKAN                       = 0000000F          INT$_STKG                        = 0000002A
FLG$V_SPECOP                     = 00000022          INT$_STKL                        = 0000002B
FLG$V_SPLALL                     = 0000001A          INT$_STKPC                       = 0000002C
FLG$V_STOIMF                     = 00000012          INT$_STKS                        = 0000002D
FLG$V_SYM2COL                    = 0000002A          INT$_STOB                        = 00000034
```

```
                                                         M  1
MACSACTONE                       ONCE-ONLY ACTION ROUTINES               16-SEP-1984 02:19:08  VAX/VMS Macro V04-00        Page  19
Symbol table                                                              5-SEP-1984 01:46:56  [MACRO.SRC]ACTONE.MAR;1           (11)

INT$_STOL                      = 0000002E                 MAC$GT_SCB                     ********   X    03
INT$_STOW                      = 00000035                 MAC$GW_LST_INST                ********   X    03
INT$_STRB                      = 0000002F                 MAC$GW_LST_LINE                ********   X    03
INT$_STRL                      = 00000031                 MAC$INSUSRSYMTB                ********   X    03
INT$_STRSB                     = 00000032                 MAC$INTOUT_2_LW                ********   X    03
INT$_STRSW                     = 00000033                 MAC$INTOUT_N                   ********   X    03
INT$_STRW                      = 00000030                 MAC$INTOUT_X                   ********   X    03
INT$_STSB                      = 00000036                 MAC$K_HD_SIZE                  ********   X    03
INT$_STSW                      = 00000037                 MAC$LIST_RAB                   ********   X    03
INT$_SUB                       = 0000000A                 MAC$OUTFRAME                   ********   X    03
INT$_SUME                      = 00000039                 MAC$PASS1_END                  ********   X    03
INT$_WRN                       = 00000038                 MAC$SET_NEW_LSB                ********   X    03
INT$_XOR                       = 0000000B                 MAC$SET_PC                     ********   X    03
LIST                             000002FB RG    03        MAC$SKIPSP                     ********   X    03
LIST_NLIST                       00000309 R     03        MAC$SKP_OPR                    ********   X    03
LST$G_DIRLIST                    ********   X   03        MAC$SRCUSRSYMTB                ********   X    03
LST$G_LONGNAMES                  ********   X   03        MAC$SRC_LIST                   ********   X    03
LST$K_BUFSIZ                   = 00000086                 MAC$SYMSCNUP                   ********   X    03
LST$K_L_P_PAGE                 = 0000003C                 MAC$_ADRLSTSYNX              = 007D9002
LST$K_TITLE_SIZ                = 00000028                 MAC$_ASGNMNTSYN              = 007D9022
MAC$AB_HD_NEWPG                  ********   X   03        MAC$_BADENTRY                = 007D902A
MAC$AB_HD_TITLE                  ********   X   03        MAC$_BLKDIRSYNX              = 007D9042
MAC$AB_HD_TSTRG                  ********   X   03        MAC$_DATALSTSYN              = 007D9062
MAC$AB_IDENT                     ********   X   03        MAC$_DIRSYNX                 = 007D906A
MAC$AB_LINEBF                    ********   X   03        MAC$_ILLASCARG               = 007D90B2
MAC$AB_TITLE                     ********   X   03        MAC$_ILLDFLTARG              = 007D90CA
MAC$AB_TMPSYM                    ********   X   03        MAC$_ILLEXPR                 = 007D90D2
MAC$AB_TOC_MSG                   ********   X   03        MAC$_ILLOPDEF                = 007D910A
MAC$CLOSE_LIST                   ********   X   03        MAC$_ILLSYMLEN               = 007D8820
MAC$DEC_OUT_L2X                  ********   X   03        MAC$_MCHINSTSYN              = 007D914A
MAC$DEC_OUT_R2L                  ********   X   03        MAC$_MSGCMAIIF               = 007D9152
MAC$DFLT_LIST                    ********   X   03        MAC$_NOTENABOPT              = 007D9172
MAC$DSPL_ARGS                    ********   X   03        MAC$_NOTINMACRO              = 007D918A
MAC$ERRORLN                      ********   X   03        MAC$_NOTLGLISOP              = 007D9192
MAC$ERRORPT                      ********   X   03        MAC$_OPRNDSYNX               = 007D91A2
MAC$ERRORPX                      ********   X   03        MAC$_REGOPSYNX               = 007D91CA
MAC$ERR_PUT                      ********   X   03        MAC$_UNDEFXFRAD              = 007D921A
MAC$GB_MODE                      ********   X   03        MAC$_UNRECSTMT               = 007D9222
MAC$GB_REG                       ********   X   03        MAC$_UNTERMARG               = 007D922A
MAC$GETCHR                       ********   X   03        MAC$_UNTERMCOND              = 007D9232
MAC$GL_DFPC_DSP                  ********   X   03        MAC_SUBSYS                   = 0000007D
MAC$GL_DIRFLG                    ********   X   03        NLIST                          00000303 RG    03
MAC$GL_DSLISF                    ********   X   03        OBJ$K_BUFSIZ                 = 00000200
MAC$GL_ENLISF                    ********   X   03        OPF$M_LASTOPR                = 00002000
MAC$GL_IF_LEVEL                  ********   X   03        OPF$M_OPTEXP                 = 00001000
MAC$GL_INTWRNPT                  ********   X   03        OPF$V_LASTOPR                = 0000000D
MAC$GL_LINEIN                    ********   X   03        OPF$V_OPTEXP                 = 0000000C
MAC$GL_LINEPT                    ********   X   03        PSC$B_NAME                     00000004
MAC$GL_LINE_CNT                  ********   X   03        PSC$B_SEG                      0000000C
MAC$GL_LIST_IT                   ********   X   03        PSC$B_UNUSED                   0000000B
MAC$GL_LIST_LVL                  ********   X   03        PSC$K_BLKSIZ                   00000013
MAC$GL_LIST_PTR                  ********   X   03        PSC$K_NO_OPTNS               = 0000000A
MAC$GL_LN_PAGE                   ********   X   03        PSC$L_CURLOC                   0000000F
MAC$GL_PC                        ********   X   03        PSC$L_LINK                     00000000
MAC$GL_SRCPAG                    ********   X   03        PSC$L_MAXLGTH                  00000005
MAC$GL_TTX_SIZ                   ********   X   03        PSC$M_ABS                    = FFFFFFF7
MAC$GL_XFRADR                    ********   X   03        PSC$M_ALIGNFLG               = 00004000
```

```
MAC
Sym

SYM
SYM
SYM
SYM
TAB
VB
VD
VF
VG
VH
VL
VO
VQ
VW
W
WB
WD
WF
WG
WH
WL
WO
WQ
WW
X1
X2

PSE
----
. A
. BL
$AB
MAC
MAC

Pha
----
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
276
```

```
                                                              N  1
MAC$ACTONE                              ONCE-ONLY ACTION ROUTINES              16-SEP-1984 02:19:08  VAX/VMS Macro V04-00        Page  20
Symbol table                                                                    5-SEP-1984 01:46:56  [MACRO.SRC]ACTONE.MAR;1           (11)

PSC$M_ALLOPTNS                   = 000003FF                SIZ...                        = 00000001
PSC$M_BYTE                       = 00004000                STB$K_PG_MISS                 = 0000000A
PSC$M_CON                        = FFFFFFFB                SUM_B_FLAGS                     0000001C
PSC$M_DEFAULT                    = 000001C8                SUM_K_BLN                       0000001D
PSC$M_EXE                        = 000000C0                SUM_L_ISDATA                    00000004
PSC$M_GBL                        = 00000010                SUM_L_STS                       00000000
PSC$M_LCL                        = FFFFFFEF                SUM_M_AUDIT                   = 00000001
PSC$M_LIB                        = 00000002                SUM_M_AUDITNEW                = 00000002
PSC$M_LONG                       = 00004800                SUM_M_DELETE                  = 00000010
PSC$M_NOEXE                      = FFFFFFBF                SUM_M_SRCUPD                  = 00000004
PSC$M_NOPIC                      = FFFFFFFE                SUM_M_SUBCLSH                 = 00000008
PSC$M_NORD                       = FFFFFF7F                SUM_Q_AUDDS                     00000008
PSC$M_NOSHR                      = FFFFFFDF                SUM_Q_FILESP                    00000010
PSC$M_NOVEC                      = FFFFFDFF                SUM_V_AUDIT                   = 00000000
PSC$M_NOWRT                      = FFFFFEFF                SUM_V_AUDITNEW                = 00000001
PSC$M_OVR                        = 00000004                SUM_V_DELETE                  = 00000004
PSC$M_PAGE                       = 00006400                SUM_V_SRCUPD                  = 00000002
PSC$M_PIC                        = 00000001                SUM_V_SUBCLSH                 = 00000003
PSC$M_QUAD                       = 00004C00                SUM_W_INSERT_NO                 0000001A
PSC$M_RD                         = 00000080                SUM_W_LINE_NO                   00000018
PSC$M_REL                        = 00000008                SYM$B_NAME                      00000004
PSC$M_SHR                        = 00000020                SYM$B_SEG                       0000000C
PSC$M_USR                        = FFFFFFFD                SYM$B_TOKEN                     0000000B
PSC$M_VEC                        = 00000200                SYM$K_BLKSIZ                    0000000D
PSC$M_WORD                       = 00004400                SYM$K_MAXLEN                  = 0000001F
PSC$M_WRT                        = 00000180                SYM$K_TWOCOL                  = 00000010
PSC$S_ALIGNMENT                  = 00000004                SYM$L_LINK                      00000000
PSC$V_ALIGNFLG                   = 0000000E                SYM$L_VAL                       00000005
PSC$V_ALIGNMENT                  = 0000000A                SYM$M_ABS                     = 00000010
PSC$V_EXE                        = 00000006                SYM$M_ASN                     = 00000100
PSC$V_GBL                        = 00000004                SYM$M_CRFO                    = 00002000
PSC$V_LIB                        = 00000001                SYM$M_DEBUG                   = 00000020
PSC$V_OVR                        = 00000002                SYM$M_DEF                     = 00000001
PSC$V_PIC                        = 00000000                SYM$M_DELMAC                  = 00000200
PSC$V_RD                         = 00000007                SYM$M_EPT                     = 00000200
PSC$V_REL                        = 00000003                SYM$M_EXTRN                   = 00000008
PSC$V_SHR                        = 00000005                SYM$M_GLOBL                   = 00000004
PSC$V_VEC                        = 00000009                SYM$M_LOCAL                   = 00000040
PSC$V_WRT                        = 00000008                SYM$M_ODBG                    = 00000400
PSC$W_FLAG                         00000009                SYM$M_REF                     = 00000080
PSC$W_OPTIONS                      0000000D                SYM$M_RELPSECT                = 00000800
RAB$L_RBF                        = 00000028                SYM$M_SUPR                    = 00004000
RAB$W_RSZ                        = 00000022                SYM$M_WEAK                    = 00000002
RDX$V_BINARY                     = 00000000                SYM$M_XCRF                    = 00001000
RDX$V_DECIMAL                    = 00000002                SYM$V_ABS                     = 00000004
RDX$V_DOUBLE                     = 00000005                SYM$V_ASN                     = 00000008
RDX$V_FLOAT                      = 00000004                SYM$V_CRFO                    = 0000000D
RDX$V_GFLOAT                     = 00000006                SYM$V_DEBUG                   = 00000005
RDX$V_HEX                        = 00000003                SYM$V_DEF                     = 00000000
RDX$V_HFLOAT                     = 00000007                SYM$V_DELMAC                  = 00000009
RDX$V_OCTAL                      = 00000001                SYM$V_EPT                     = 00000009
REG$_PC                          = 0000000F                SYM$V_EXTRN                   = 00000003
SBTTL                              00000156 RG     03      SYM$V_GLOBL                   = 00000002
SBT_PUT_LIN                        00000268 R      03      SYM$V_LOCAL                   = 00000006
SEMI                             = 0000003B                SYM$V_ODBG                    = 0000000A
SETDFL                             000003F8 RG     03      SYM$V_REF                     = 00000007
SET_XCRF                           000003EC R   D  03      SYM$V_RELPSECT                = 0000000B
```

```
MAC1
VAX-

Ther
231
14 p

Macr
----
_$2
-$2
TOTA

537

Ther

MACR
```

```
SYM$V_SUPR                      = 0000000E
SYM$V_WEAK                      = 00000001
SYM$V_XCRF                      = 0000000C
SYM$W_FLAG                        00000009
SYS$PUT                           ********  GX    03
TAB                             = 00000009
TITLE                             000000EE RG     03
X                               = 00000010
X1                              = 00000400
X2                              = 0000000F

                                               +----------------+
                                               ! Psect synopsis !
                                               +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
. BLANK .                       00000000  (     0.)  01 (   1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$ABS$                           0000001D  (    29.)  02 (   2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
MAC$RO_CODE_P1                  00000499  (  1177.)  03 (   3.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD  NOWRT NOVEC LONG

                                           +------------------------+
                                           ! Performance indicators !
                                           +------------------------+

Phase                       Page faults    CPU Time        Elapsed Time
-----                       -----------    --------        ------------
Initialization                       29    00:00:00.02     00:00:02.32
Command processing                  121    00:00:00.44     00:00:03.08
Pass 1                              270    00:00:04.92     00:00:22.15
Symbol table sort                     0    00:00:00.61     00:00:02.33
Pass 2                              132    00:00:01.32     00:00:05.61
Symbol table output                  55    00:00:00.24     00:00:01.52
Psect synopsis output                 1    00:00:00.02     00:00:00.02
Cross-reference output                0    00:00:00.00     00:00:00.00
Assembler run totals                610    00:00:07.58     00:00:37.04

The working set limit was 1500 pages.
45183 bytes (89 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 649 non-local and 50 local symbols.
630 source lines were read in Pass 1, producing 24 object records in Pass 2.
24 pages of virtual memory were used to define 22 macros.

                                          +--------------------------+
                                          ! Macro library statistics !
                                          +--------------------------+

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SHRLIB]SUM.MLB;1                          3
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1                    12
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      7
TOTALS (all libraries)                                22

754 GETS were required to define 22 macros.
```

```
There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ACTONE/OBJ=OBJ$:ACTONE MSRC$:ACTONE/UPDATE=(ENH$:ACTONE)+LIB$:MACRO/LIB+SHRLIB$:SUM/LIB
```

0223 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DEFINE
MAR

LOGINCMD
LIS

ACTONE
LIS

MESSAGES
LIS

ACTCHR
LIS

VALIDATE
LIS

ACTIF
LIS

PROTCLI
LIS

CRFMACROS
MAR

MACRO

MACRO32
MAP

0224 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

ACTPRI
LIS

ARGSCN
LIS

BDYSCN
LIS

CRFSUB
LIS

ACTOPC
LIS

ACTSTA
LIS

APSECT
LIS

CRFDAT
LIS

ACTREP
LIS

COMPUT
LIS